Practica 4

1) ¿Qué servicios presta la capa de red? ¿Cuál es la PDU en esta capa?

La capa de Red permite la conexión desde un host origen a un host destino. En TCP/IP está implementada en el protocolo **IP**, e interviene en cada host y encaminador intermedio (**router**)

El PDU es el **datagrama**, que encapsula los segmentos de **transporte** agregándole las direcciones IP origen y destino entre otras cosas.

Ofrece dos tipos de servicios:

* **Retransmisión** **(forwarding,encaminamiento)**: Los datagramas que llegan a un router son dirigidos a la interfaz de salida apropiada.
* **Encaminamiento** **(routing,rutado)**: Determina el camino que toman los paquetes desde el origen al destino (según diferentes algoritmos de routeo).

Otro tipo de servicio, aunque ausente en Internet, es el establecimiento de conexión (en redes tipo ATM -de conmutación de circuitos, o circuitos virtuales-): los routers intermedios del camino establecen la conexión virtual, reservando recursos -bandwidth, buffers...- y "recordando" el camino elegido antes que los paquetes se comiencen a transmitir (agregan entrada a tabla de ruteo).

2) Compare los siguientes modelos de servicios de red:

|  |  |  |
| --- | --- | --- |
|  | **Datagrama** | **Circuitos Virtuales** |
| **¿Todos los paquetes siguen el mismo camino?** | **NO** | **SI** |
| **¿Cuenta con una fase de establecimiento y otra de cierre de circuito?** | **NO** | **SI** |
| **¿Usa mensajes de señalización?** | **NO** | **SI** -para establecimiento/mantención/cierre de conexión- |
| **¿Usa tablas de enrutamiento?** | **SI** | **SI** |

3) ¿Qué dispositivo es considerado sólo de esta capa? Explique las dos funciones principales que debe realizar .  
  
El dispositivo de capa de **Red** es el **router**. Acorde a los dos servicios de la capa de Red, este dispositivo debe:

* Ejecutar algoritmos/protocolos de enrutamiento que seleccionen hacia dónde reenviar un datagrama recibido.
* Encaminar/conmutar los datagramas que llegan a una interfaz o puerto de entrada, a la interfaz o puerto de salida seleccionada por el algoritmo. La conmutación puede hacerse vía memoria -control directo de una CPU-, vía bus compartido en el router o vía *crossbar* o red de interconexión.

El puerto de salida puede realizar *buffering* si su tasa de transmisión es inferior a la de llegada de datos desde el entramado de conmutación (produciendo un retraso), si el buffer se llena, pueden perderse paquetes. Del mismo modo si la tasa de llegada de datagramas al puerto de entrada es superior a la velocidad de conmutación, el puerto de entrada utiliza *buffering*.

4) En las redes IP el ruteo puede hacerse en forma estática como dinámica. Describa

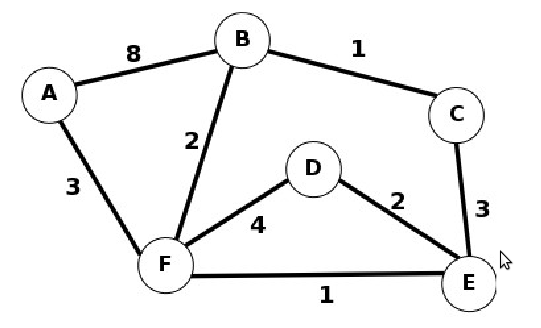
conceptualmente como funciona cada uno de ellos e indique ventajas y desventajas de cada método.   
  
Ruteo estático o dinámico: es una clasificación de los algoritmos de ruteo. En el ruteo estático, las rutas cambian muy lentamente a lo largo del tiempo, normalmente como resultado de una intervención humana (porque la persona edita manualmente las tablas de encaminamiento).   
Los algoritmos de rutado dinámico cambian los caminos de rutado según cambia la carga del trafico o la topología de la red. Un algoritmo dinámico podrá ejecutarse bien periódicamente, o bien en respuesta directa a un cambio en la topología o en los costes de los enlaces.

5) Los algoritmos de ruteo dinámico se dividen en estado enlace y vector distancia. Dado el siguiente cuadro compare:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **¿Cada router conoce la topología de la red completa?** | **¿Converge rápidamente?** | **Protocolos que lo implementan** |
| **Vector distancia** | No | No | RIP |
| **Estado de enlace** | Si | Si | OISPF |

6) Dado el apunte que explica en detalle los algoritmos de ruteo dinámicos de estado enlace y vector distancia, y dado el siguiente grafo, indique el procedimiento para calcular el camino de costo mínimo a partir del nodo B según los siguientes cuadros:

Complete el cuadro utilizando el algoritmo de estado enlace:



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Iteracion** | **N** | **D(A)p(A)** | **D(C)p(C)** | **D(D)p(D)** | **D(E)p(E)** | **D(F)p(F)** |
| **0** | B | 8,B | 1,B |  |  | 2,B |
| **1** | BA |  |  |  |  |  |
| **2** |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |
| **5** |  |  |  |  |  |  |

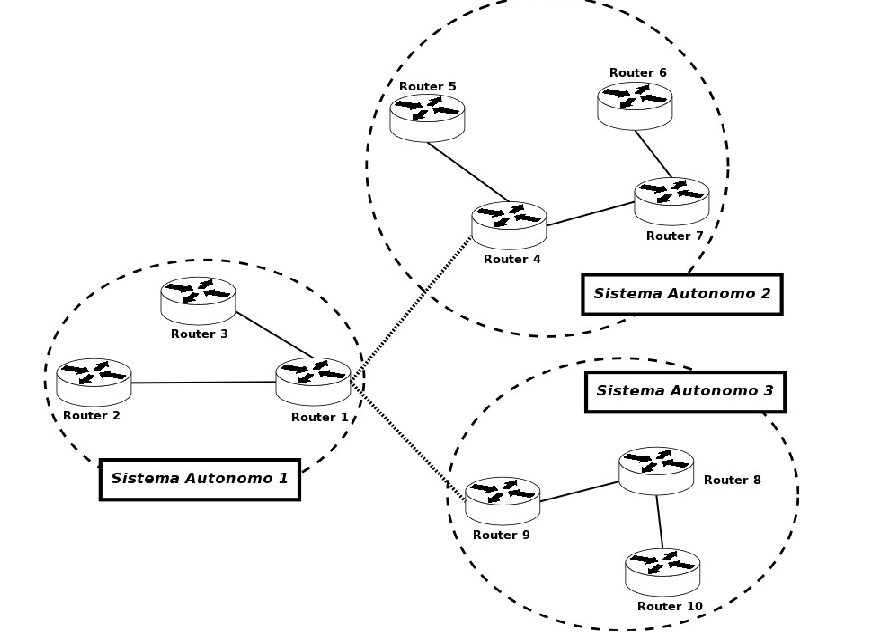
|  |  |  |  |
| --- | --- | --- | --- |
| **DB( \_ , \_ )** | **A** | **F** | **C** |
| **A** | 8 | **5** | 7 |
| **C** | 14 | 5 | **1** |
| **D** | 14 | 6 | **5** |
| **E** | 12 | **3** | 4 |
| **F** | 11 | **2** | 4 |

7) ¿Qué son los sistemas autónomos y por qué resultan necesarios?

Un Sistema Autónomo (en inglés, Autonomous System: AS) es un conjunto de redes y dispositivos router IP que se encuentran administrados por una sola entidad (o en algunas ocasiones varias) que cuentan con una política común de definición de trayectorias para Internet.   
Los Sistemas Autónomos se comunican entre sí mediante routers BGP y se intercambian el tráfico de Internet que va de una red a la otra. A su vez cada Sistema Autónomo es como una Internet en pequeño, ya que su rol se llevaba a cabo por una sola entidad, típicamente un Proveedor de Servicio de Internet (ISP) o una gran organización con conexiones independientes a múltiples redes, las cuales se apegaban a una sola y clara política de definición de trayectorias definida.   
Técnicamente un Sistema Autónomo se define como “un grupo de redes IP que poseen una política de rutas propia e independiente”. Esta definición hace referencia a la característica fundamental de un Sistema Autónomo: realiza su propia gestión del tráfico que fluye entre él y los restantes Sistemas Autónomos que forman Internet. Aún considerando que el ISP podía soportar múltiples sistemas autónomos, Internet solo considera la política de definición de trayectorias establecida por el ISP. Por lo tanto, el ISP debería contar con un ASN registrado. Un número de AS o ASN se asigna a cada AS para ser utilizado por el esquema de encaminamiento BGP, este número identifica de manera única a cada red dentro del Internet.  
Los protocolos de ruteo que utilizan internamente se denominan **IGP** (Interior Gateway Protocol), y pueden ser: RIP, IGRP, EIGRP, OSPF, IS-IS...  
Los SA permiten el **ruteo jerárquico**. Cada SA se conecta a un *router de borde* o *gateway*, que lo conecta a otros. Se denominan **EGP** (Exterior Gateway Protocols) a los protocolos entre distintos AS (GGP, EGP, BGP).  
El **ruteo jerárquico** permite salvar los problemas que supondrían utilizar los mismos protocolos entre diferentes subredes:

* **Escala:** El enorme **tamaño** que alcanzarían las tablas de ruteo, el **overhead** de intercambiar información entre los routers para una red de grandes dimensiones y el enorme **tiempo de convergencia** que se tendría.
* Autonomía administrativa: La imposibilidad de elegir un protocolo específico por parte de los administradores de las redes.

8) A partir del siguiente gráfico indique



¿Qué tipo de algoritmo se utiliza para compartir información entre los routers 4 y 7?   
  
IGP

¿Qué tipo de algoritmo se utiliza para compartir información entre los routers 1 y 4?   
  
EGP

¿Qué tipo de algoritmos alimentan las tablas de ruteo de los routers 3 y 10?   
  
IGP y despues EGP

¿Qué tipo de algoritmos alimentan las tablas de ruteo de los routers 1 y 4?   
  
Algoritmo de IGP

9) Dadas las siguientes direcciones IP, indique su clase y si las mismas corresponden a direcciones de máquina, de red o broadcast: 203.15.6.87, 67.154.0.0, 171.58.0.0, 127.0.0.1, 24.130.56.0, 197.54.66.255

|  |  |  |
| --- | --- | --- |
| **Dirección IP** | **Clase** | **Tipo** |
| **203.15.6.87** | C | Host/Interface |
| **67.154.0.0** | A | Host/Interface |
| **171.58.0.0** | B | Red |
| **127.0.0.1** | A | Loopback (Host/Interface) |
| **24.130.56.0** | A | Host/Interface |
| **197.54.66.255** | C | Broadcast |

**Clase A: el primer bit esta en 0. [0 .. 255-128=127] Mascara: /8 (defecto)**

**Clase B: el primero en 10.[0 .. 255-64=191] Mascara: /16**

**Clase C: 110 Mascara: /24 [0 .. 255-**

**Clase D: 1110**

Hasta 127 A; hasta 191 B; hasta 223 C; hasta 239 Multicast.

la mascara de red se usa para calcular (router) para saber a que red pertenece la IP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Clase | Rango | N° de Redes | N° de Host | [Máscara de Red](file:///C:\Users\Nicolas\Documents\GitHub\wiki\M%25C3%25A1scara_de_subred) | [Broadcast ID](file:///C:\Users\Nicolas\Documents\GitHub\wiki\Broadcast_(inform%25C3%25A1tica)) |
| A | 1.0.0.0 - 127.255.255.255 | 126 | 16.777.214 | 255.0.0.0 | x.255.255.255 |
| B | 128.0.0.0 - 191.255.255.255 | 16.382 | 65.534 | 255.255.0.0 | x.x.255.255 |
| C | 192.0.0.0 - 223.255.255.255 | 2.097.150 | 254 | 255.255.255.0 | x.x.x.255 |
| D | 224.0.0.0 - 239.255.255.255 |  |  |  |  |
| E | 240.0.0.0 - 255.255.255.255 |  |  |  |  |

* Una dirección IP no identifica un equipo (computadora/router) específico, sino que identifica una **interface** de un host en una red. Un equipo con acceso a múltiples redes (por ejemplo, un router) debe tener asignada una dirección IP por cada una de éstas.
* La dirección 0.0.0.0 es utilizada por las máquinas cuando están arrancando o no se les ha asignado dirección.
* La dirección que tiene su parte de host a cero sirve para definir la red en la que se ubica. Se denomina **dirección de red**.
* La dirección que tiene su parte de host a unos sirve para comunicar con todos los hosts de la red indicada. Se denomina **dirección de broadcast directo**.
* La dirección que tiene su parte de red y de host a unos sirve para comunciar con todos los equipos de la red local (IP 255.255.255.255). Se denomina **dirección de broadcast limitado**.
* Las direcciones 127.x.x.x se reservan para pruebas de retroalimentación. Se denomina **dirección de bucle local** o **loopback**.
* Hay ciertas direcciones en cada clase de dirección IP que no están asignadas y que se denominan **direcciones privadas**; pueden ser utilizadas por los hosts que usan *traducción de dirección de red* (**NAT**) para conectarse a una red pública o por los hosts que no se conectan a Internet. En una misma red no puede existir dos direcciones iguales, pero sí se pueden repetir en dos redes privadas que no tengan conexión entre sí o que sea a través de NAT. Las direcciones privadas son:
* Clase A: 10.x.x.x (8 bits red, 24 bits hosts)
* Clase B: 172.16.x.x a 172.31.x.x (12 bits red, 20 bits hosts)
* Clase C: 192.168.x.x (16 bits red, 16 bits hosts)

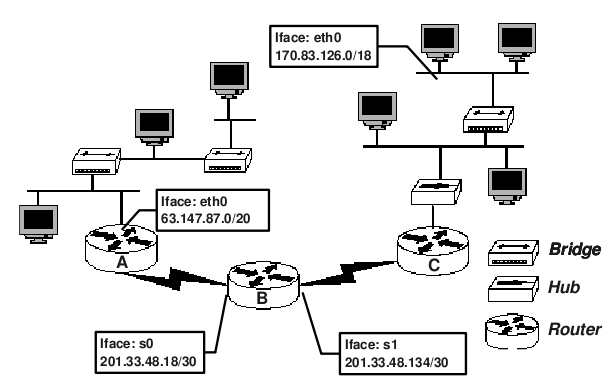
10) ¿Qué son las subredes?  
  
La división de subredes es la obtención de otras direcciones de red basadas en una sola dirección con el uso de la **máscara de subred**, "*pidiendo bits prestados*" a la parte del host/interface (dependiendo de la cantidad de bits que se pidan, será la cantidad de subredes que se creen a partir de la original).  
Una subred es un **rango de direcciones lógicas**. Cuando una red se vuelve muy grande, conviene dividirla en subredes, para reducir el tamaño de los dominios de broadcast, y hacer la red más manejable.  
Típicamente los routers constituyen los límites entre las subredes. La comunicación desde/hasta otras subredes es hecha mediante un router específico. Sin embargo, las subredes permiten dividir lógicamente una red a pesar de su diseño físico, pudiéndose dividir en varias subredes configurando diferentes host que utilicen diferentes routers. La dirección de todos los nodos en una subred comienzan con la misma secuencia binaria, que es su ID de red e ID de subred (en IPv4, las subredes deben ser identificadas por la **base** de la dirección y una **máscara de subred**).  
Las subredes simplifican el enrutamiento, representándose típicamente cada una como una fila en las tablas de ruteo en cada router conectado. Fueron usadas antes de IPv4 para permitir a una red grande tener un número importante de redes más pequeñas dentro, controladas por varios routers.  
Permiten el Enrutamiento Interdominio Sin Clases (**CIDR**).  
Los últimos dos bits del último octeto (los menos significativos) nunca se asignan a la subred, sea cual sea la clase de dirección IP. **Como consecuencia, una dirección de subred jamás terminará en un número impar**. Por otra parte, el uso de todos los bits disponibles para crear subredes dará como resultado subredes con sólo dos Hosts utilizables (un método práctico de conservación de direcciones para el direccionamiento de enlace *punto a punto*, donde no existe otro direccionamiento más que los dos enlaces conectado entre sí).  
  
11) Dada la red 195.200.45.0/24. Se necesitan definir 9 subredes. Indique la máscara utilizada y las nueve primeras subredes. Luego tome una de ellas e indique el rango de direcciones asignables en esa subred, dirección de red y broadcast.

Subred: /28. 4 bits → 16 subredes.

SubRed  
nº1 195.200.45. 0000 | 0000   
nº2 195.200.45. 0001 | 0000  
nº3 195.200.45. 0010 | 0000  
nº4 195.200.45. 0011 | 0000

2⁴ = 16 hosts, empezando desde el 0 → 15 host. 15 – 2 (red y broadcast) 13 hosts utilizables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dirección IP (subred)** | **Mascara** | **Rango direcciones** | **Dirección Red** | **Dirección Broadcast** |
| 195.200.45.0 | 28 | 195.200.45.1 – 195.200.45.14 | 195.200.45.0 | 195.200.45.15 |
| 195.200.45.16 | 28 | 195.200.45.17 – 195.200.45.30 | 195.200.45.16 | 195.200.45.31 |
| 195.200.45.32 | 28 | 195.200.45.33 – 195.200.45.46 | 195.200.45.32 | 195.200.45.47 |
| 195.200.45.48 | 28 | 195.200.45.49 – 195.200.45.62 | 195.200.45.48 | 195.200.45.63 |
| 195.200.45.64 | 28 |  |  |  |

  
12) Dado el siguiente gráfico complete

1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Interfaze** | **IP** | **Direccion de red – Clase** | **Subred – Mascara** | **Direccion broadcast** | **Dir utilizables (subred)** |
| **eth0** | 170.83.126.0/18 | 170.83.0.0 – Clase B | 170.83.64.0/18 | 170.83.127.255 | 16382 |
| **eth0' (A)** | 63.147.87.0/20 | 63.0.0.0 – Clase A | 63.147.80.0/20 | 63.147.95.255 | 4094 |
| **s0** | 201.33.48.18/30 | 201.33.48.0 – Clase C | 201.33.48.16/30 | 201.33.48.19 | 2 |
| **S1** | 201.33.48.134/30 | 201.33.48.0 – Clase C | 201.33.48.132/30 | 201.33.48.135 | 2 |

Eth0: 170.83. 01111110 . 00000000 → dir IP  
 170.83. 11000000 . 00000000 → mascara /18  
-----------------------------------------------------  
 170.83. 01000000 . 00000000 = 64!  
  
  
  
  
eth0': 63.147. 01010111 . 00000000 → dir IP  
 63.147. 11110000 . 00000000 → mascara /20  
------------------------------------------------------  
 63.147. 01010000 . 00000000 = 80!

s1: 201.33.48.134/30 → 134: 100001 | 10 → host, la otra es 01. (00 y 11 red y broadcast)  
 201.33.48. 10000110 → IP  
 201.33.48. 11111100 → mascara /30  
----------------------------------------------  
 201.33.48. 100001 00 2 bits de host  
  
s0: 201.33.48.18/30 → 10010 | 10 → la otra es 01.

2. Escoja una dirección IP adecuada para cada una de las interfaces de cada uno de los routers.

Con /30 tengo 2 direcciones utilizables (4 en realidad con 2 bits, pero una de red y una de broadcast) asique es la 201.33.48.133/30 (router C), la 201.33.48.17/30 (router A) y la 170.83.127.0/18 (router C del lado de arriba) bien??  
  
3. Defina una tabla de ruteo para cada router en el gráfico, de forma tal de que todos los dispositivos en la red puedan comunicarse   
  
Destino: direccion de red conectada a la interfaz.   
Pasarela: a que interfaz (entonces dir ip) ir para la red indicada en destino  
  
Tabla router A:

|  |  |  |  |
| --- | --- | --- | --- |
| **Destino** | **Pasarela (gateway)** | **Mascara** | **Interfaz** |
| 201.33.48.16/30 | \* | 255 . 255 . 255 . 252 | s0 |
| 63.147.80.0/20 | \* | 255.255.240.0 | eth0 |
| 0.0.0.0 (default gw) | 201.33.48.18/30 | 255 . 255 . 255 . 252 | s0 |
|  |  |  |  |

63.147. 0101 | 0111 . 0

Tabla router B:

|  |  |  |  |
| --- | --- | --- | --- |
| **Destino** | **Pasarela(gateway)** | **Mascara** | **Interfaz** |
| 201.33.48.16/30 | \* | 255 . 255 . 255 . 252 | s0 |
| 201.33.48.132/30 | \* | 255 . 255 . 255 . 252 | s1 |
| 63.147.80.0/20 | 201.33.48.17/30 | 20 | s0 |
| 170.83.64.0/18 | 201.33.48.133/30 | 18 | s1 |

170.83.126.0/18 =   
170.83. 01 | 111110 . 0 → subred: 170.83.64.0/18  
  
Tabla router C:

|  |  |  |  |
| --- | --- | --- | --- |
| **Destino** | **Pasarela(gateway)** | **Mascara** | **Interfaz** |
| 170.83.64.0 | \* | 18 | s0 |
| 201.33.48.132/30 | \* | 30 | s1 |
| 0.0.0.0 (default gw) | 201.33.48.134 | 30 | s1 |

13. Dada la IP 65.0.0.0/8. Se necesitan definir 934 subredes. Indique que máscara deberia ser utilizada. Indique cual seria la subred numero 817 indicando el rango de direcciones asignables, dirección de red y broadcast.

2¹⁰=1024, asique 10 bits mas para definir las subredes. Mascara /18  
65.0.0.0/18  
  
11001100 (segundo octeto) . 01 (tercer octeto) = 817  
65.204.64.0/18 → sub red 817  
16382 direcciones utilizables (2¹⁴ – 2)  
broadcast → 65.204.127.255  
 65.294. 01111111 . 11111111

14) Dada la dirección IP 172.16.58.223/26. ¿ Cuál es la dirección de subred? ¿Y la de broadcast?  
  
Tenes 26 bits de red, 6 de hosts.  
  
Subred: 172.16.58.192  
Broadcast: 172.16.58.255

15) *CIDR achica las tablas de ruteo, un router en vez de publicar sus (128 por ej) redes, publica una sola con otra mascara.*

**Classless Inter-Domain Routing** (**CIDR** Encaminamiento Inter-Dominios sin Clases) es una mejora en el modo como se interpretan las direcciones IP. Su introducción permitió una mayor flexibilidad al dividir rangos de direcciones IP en redes separadas, reemplazando la sintaxis previa para nombrar direcciones IP, las **clases** de redes.  
CIDR usa la técnica **VLSM** (**Variable-Length Subnet Masking** - Máscara de Subred de Longitud Variable), para hacer posible la asignación de prefijos de longitud arbitraria (la división red/host puede ocurrir en cualquier bit de los 32 que componen la dirección IP).  
Un gran beneficio de CIDR es la posibilidad de agregar prefijos de encaminamiento -**supernetting**-. Por ejemplo, dieciséis redes /24 contíguas pueden ser agregadas y publicadas en los routers como una sola ruta /20 (si los primeros 20 bits de sus respectivas redes coinciden). Dos redes /20 contiguas pueden ser agregadas en una /19, etc. Esto permite reducir significativamente el número de rutas que los routers tienen que conocer (reduciendo memoria, recursos, etc.) y previene una explosión de tablas de encaminamiento.  
Decimos que una dirección IP **está incluida** en un **bloque CIDR**, y que **encaja** con el prefijo CIDR, si los N bits iniciales de la dirección y el prefijo son iguales. Por tanto, para entender CIDR es necesario visualizar la dirección IP en binario.  
  
  
16) 212.56.132.0/24 → 1000 0100  
 212.56.133.0/24 1000 0101  
 212.56.134.0/24 1000 0110  
 212.56.135.0/24 1000 0111  
  
Mando 212.56.132.0/22  
  
  
17) Listar las redes que involucra el bloque CIDR 200.56.168.0/21.  
  
200.56. 168 . 0  
200.56. 10101 | 000 . 00000000  
Subred /21  
  
200.56.10101 | 000 . 00000000 = 200.56.168.0/24  
200.56.10101 | 001 . 00000000 = 200.56.169.0/24  
200.56.10101 | 010 . 00000000 = 200.56.170.0/24  
200.56.10101 | 011 . 00000000 = 200.56.171.0/24  
200.56.10101 | 100 . 00000000 = 200.56.172.0/24  
200.56.10101 | 101 . 00000000 = 200.56.173.0/24  
200.56.10101 | 110 . 00000000 = 200.56.174.0/24  
200.56.10101 | 111 . 00000000 = 200.56.175.0/24  
  
  
18) Listar las redes que involucra el bloque CIDR 195.24.0.0/13 o 195.24/13.  
  
195. 00011 | 000 . 00000000 . 000000000 = 195.24.0.0/24  
195. 00011 | 001 . 00000000 . 000000000 = 195.25.0.0/24  
….. **2¹¹ sub redes**  
  
  
19) Una máquina que se conecta a Internet, ¿tiene tabla de ruteo?  
  
Si, es la que se ve con el comando 'route' o 'netstat -r'.  
  
  
20) El comando netstat presentado en la práctica anterior, al igual que el comando route permite visualizar las tablas de ruteo. Analice su salida evaluando que información es vital para el proceso de enrutamiento.   
(con -n te muestra en numeros las direcciones en vez de su nombre simbolico)  
  
0.0.0.0 → default

**frank@frank-laptop ~ $ route   
Tabla de rutas IP del núcleo   
Destino Pasarela Genmask Indic Métric Ref Uso Interfaz**

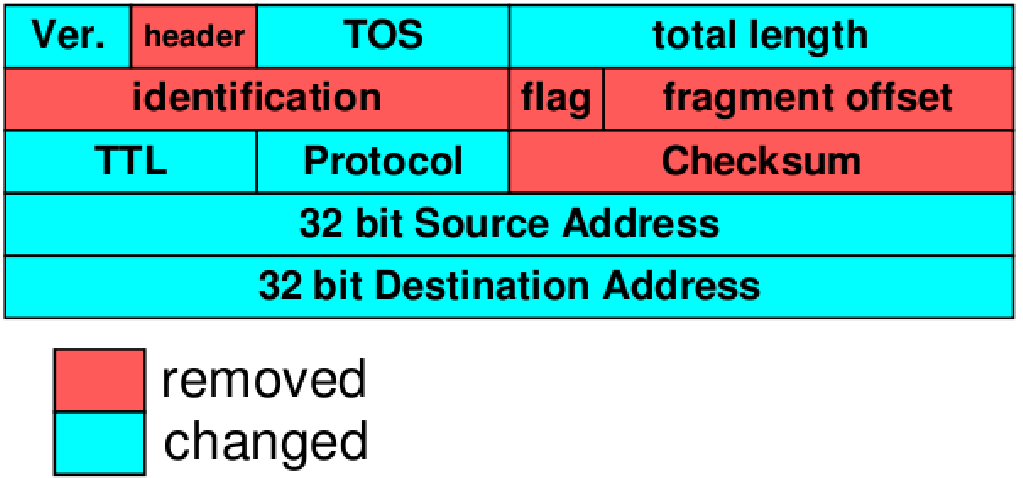
190.189.169.0 0.0.0.0 255.255.255.0 U 1 0 0 eth1   
169.254.0.0 0.0.0.0 255.255.0.0 U 1000 0 0 eth1   
0.0.0.0 190.189.169.1 0.0.0.0 UG 0 0 0 eth1   
La primer columna indica *dirección de red* (debe interpretarse de acuerdo a la *máscara* de la tercera columna) y la segunda columna indica el *gateway* -router- a utilizar (0.0.0.0 o \* se utiliza para indicar que hay conexión directa a la red en cuestión (default)).  
La columna *Flag*s(indic) indica las características de la ruta:

U (route is **u**p)  
 H (target is a **h**ost)  
 G (use **g**ateway)  
 R (**r**einstate route for dynamic routing)  
 D (**d**ynamically installed by daemon or redirect)  
 M (**m**odified from routing daemon or redirect)  
 A (installed by **a**ddrconf)  
 C (**c**ache entry)  
 ! (reject route)

**Metric**: Valor utilizado para cuantificar la ruta. IP utiliza este valor para seleccionar la mejor de dos o más rutas alternativas a la misma red.  
**Ref**: Cantidad de veces que esta ruta fue utilizada para establecer una conexión.   
**Uso**: Cantidad de paquetes trasmitidos a través de esa ruta.  
**Interfaz**: Interfaz de salida a la ruta.  
La salida de netstat agrega las columnas: *MSS* -Maximum Segment Size-, *Window* -tamaño de la ventana de TCP para las conexiones hacia esa ruta- e *irtt* -initial round trip time para conexiones TCP, en milisegundos.  
  
  
21) Describa qué hacen los comandos ping y traceroute (tracert en windows).  
  
**ping -bfr -c *count* -i *interval -*I *interface|adress -*s *packetsize* -t *ttl* -w *deadline* -W *timeout* IPDestino**  
Utiliza el protocolo ICMP para solicitar a un host/gateway una respuesta de *echo*. Envía un datagrama ECHO\_REQUEST al IP indicado. Útil para controlar que una interfaz este funcionando o que hay conectividad. Al terminar, muestra algunas estadísticas como cantidad de paquetes perdidos, TTL mínimo/medio/máximo...  
-b Permite hacer ping a una dirección de broadcast.  
-c *count* Se detiene luego de enviar *count* ECHO\_REQUEST.  
-f Imprime un . por cada ECHO\_REQUEST y un backspace (borra el punto) por cada ECHO\_REPLY recibido, mostrando cuantos paquetes se pierden  
-i *interval* Permite indicar el intervalo a esperar, en segundos, entre cada paquete.  
-I *interface|address* Permite indicar la interfaz/IP que se usará como dirección de origen.  
-R Incluye la opción Record route en el paquete ECHO\_REQUEST y muestra el buffer de ruta de los paquetes (el encabezado IP tiene tamaño suficiente para sólo 9 rutas).  
-n Mostrar direcciones númericas, no intentar obtener nombres de host.  
-r Permite saltear las tablas de ruteo normal y enviar el paquete a un equipo en una interfaz adjunta.  
-s *packetsize* Cantidad de bytes de datos a transmitir.  
-t *ttl*  Permite indicar el TTL de IP -time to live-.  
-w *deadline* Permite indicar un tiempo máximo de funcionamiento. Ping terminará pasado ese tiempo sin importar cuántos paquetes haya enviado o recibido. Si se usa junto a -c, ping terminará al cumplirse el tiempo o al enviar la cantidad indicada de paquetes, lo que ocurra primero.  
-W *timeout* Cantidad de segundos que ping esperará una respuesta. Por defecto, dos RTTs.  
**traceroute -4|6 -ITFn -f *firstTTL* -m *maxTTL* -N *squeries* -p *port* -q *nqueries* host**Muestra la ruta que siguen los paquetes por la red hasta llegar a un destino.  
Utiliza el campo TTL de IP, enviando paquetes con valores incrementales desde TTL=1, y recibiendo las respuestas ICMP de tipo TIME\_EXCEEDED de cada gateway/router en el camino hasta el destino.  
Para cada valor de TTL envía tres paquetes, luego imprime el gateway que respondió y el RTT obtenido. Si no obtiene respuesta en 5 segundos (por defecto) para algún paquete, imprime un asterisco. Los paquetes se envían a un puerto UDP que difícilmente se utilice, de modo que al llegar al destino, éste responda un ICMP PORT\_UNREACHABLE o un segemento TCP de tipo RESET, denegando la conexión.  
-4|6 Forzar el uso de IPv4 o IPv6  
-I Utilizar ICMP ECHO para los paquetes de sondeo. Si no se indica -I ni -T, utiliza UDP.  
-T Utilizar TCP SYN para los paquetes de sondeo. Si no se indica -I ni -T, utiliza UDP.  
-F Marca el bit de "Don't Fragment" para que los routers intermedios no fragementen paquetes mayores al MTU -Maximun Transfer Unit- del enlace.  
-f *firstTTL* Indica con qué valor de TTL comenzar (por defecto, 1)  
-i *interface* Permite indicar la interfaz a utilizar para los paquetes.  
-m *maxTTL* Permite indicar la cantidad maxima de "saltos" que se intentará (por defecto, 30).  
-N *squeries* Cantidad de paquetes a enviar simultáneamente (por defecto, 16).  
-n Mostrar direcciones númericas, no intentar obtener nombres de host.  
-p *port* Para tracing con UDP/TCP indica el número de puerto destino a utilizar. Para ICMP, el número de secuencia inicial.  
-q *nqueries* Cantidad de paquetes para cada "salto" posible (cada valor TTL). Por defecto, 3.  
  
22) ¿Que es y para que sirve la dirección 127.0.0.1? ¿Que PC responde al siguiente comando: *ping 127.0.0.1*?   
  
La direccion 127.0.0.1, denominada *loopback* o *localhost*, es una direccion reservada (toda la gama 127.x.x.x tiene el mismo efecto, ya que se encuentra reservada para igual propósito) que poseen todos los dispositivos (computadoras, routers), independientemente de si disponen o no de una interfaz de red, que **apunta a si mismos**, por lo cual no llega a salir nunca por la interfaz. Sus fines pueden ser para comprobar el funcionamiento de TCP/IP haciendo *ping 127.0.0.1*, al recibir respuesta se puede asumir que el software asociado al protocolo está bien (el hardware como la tarjeta de red no lo conocemos con esta prueba, ya que no llega a salir del propioequipo).  
  
  
23) ¿Que es NAT y para que sirve? De un ejemplo de su uso y analice como funcionaria en ese entorno. Ayuda: analizar el servicio de Internet hogareño y como es posible que para una conexión a Internet existan varias computadoras que usan la conexión.   
  
**NAT** (Network Address Translation - Traducción de Dirección de Red) es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles. Consiste en convertir en tiempo real las direcciones utilizadas en los paquetes transportados.   
Una pasarela NAT cambia la dirección origen en cada paquete de salida y, dependiendo del método, también el puerto origen para que sea único (PAT). Estas traducciones de dirección se **almacenan en una tabla**, para recordar qué dirección y puerto le corresponde a cada dispositivo cliente y así saber donde deben regresar los paquetes de respuesta. Si un paquete que intenta ingresar a la **red interna** no existe en la tabla de traducciones, entonces es descartado. Debido a este comportamiento, se puede definir en la tabla que en un determinado puerto y dirección se pueda acceder a un determinado dispositivo, como por ejemplo un servidor web, lo que se denomina NAT inverso o DNAT (Destination NAT).  
NAT tiene muchas formas de funcionamiento, entre las que destacan: estatico, dinamico y de sobrecarga.  
  
En un hogar con un router y 3 maquinas conectadas a el, las mismas tienen la misma IP publica (en internet) que es la que asigna el ISP local; pero “adentro” de la LAN tienen IPs privadas (del rago reservado para direcciones privadas de cada clase A, B, C o D). Entonces el router hace NAT, esto es editar cada paquete emitido por alguna maquina, cambiando la IP origen (que viene de la LAN y es privada) por la IP publica, tambien puede hacer PAT (traduccion de puertos).

24) ¿Que especifica la RFC 1918 y como se relaciona con NAT?  
  
Especifica el rango de direcciones privadas dentro de cada clase de direcciones IP.  
  
10.0.0.0 - 10.255.255.255 (prefijo 10/8)   
172.16.0.0 - 172.31.255.255 (prefijo 172.16/12)   
192.168.0.0 - 192.168.255.255 (prefijo 192.168/16)  
  
“Nos referiremos al primer bloque como "bloque de 24 bits", al segundo como "bloque de 20 bits" y al tercero como "bloque de 16 bits". Dese cuenta que (en la notación anterior a CIDR) el primer bloque no es más que un único número de red de clase A, mientras que el segundo bloque es un conjunto de 16 números de red de clase B contiguos, y el tercer bloque es un conjunto de 256 números de red de clase C contiguos.”  
  
Esto se relaciona con NAT dandole uso a las direcciones privadas en una red local y atravez de NAT traducirlas a IP publicas.

25) ¿Que es Ipv6? Enumere diferencias existentes en el formato de datagramas respecto de Ipv4.  
  
El protocolo Internet versión 6 (IPv6) es una nueva versión de IP (Internet Protocol), definida en el RFC 2460 y diseñada para reemplazar a la versión 4 (IPv4) RFC 791, actualmente en uso dominante.  
IPv4 posibilita 4.294.967.296 (2³²) direcciones de red diferentes, un número inadecuado para dar una dirección a cada persona del planeta, y mucho menos a cada vehículo, teléfono, PDA, etcétera. En cambio, IPv6 admite 340.282.366.920.938.463.463.374.607.431.768.211.456 (2¹²⁸ o 340 sixtillones de direcciones) —cerca de 3.4 × 1020 (340 trillones de direcciones) por cada pulgada cuadrada (6.7 × 1017 o 670 mil billones de direcciones/mm2) de la superficie de La Tierra.  
  
Cambios en el datagrama con respecto a IPv4:



26)   
  
pc1:~# ifconfig eth0 192.168.1.2 netmask 255.255.255.0  
router1:~# ifconfig eth0 192.168.1.1 netmask 255.255.255.0  
router1:~# ifconfig eth1 200.0.0.1 netmask 255.255.255.0  
router1:~# ifconfig eth2 200.100.0.254 netmask 255.255.255.0  
pc3:~# ifconfig eth0 192.168.3.2 netmask 255.255.255.0  
pc2:~# ifconfig eth0 192.168.2.2 netmask 255.255.255.0  
router2:~# …  
…  
  
1. Router1 envía todo el tráfico desconocido a Router2   
2. Router2 envía todo el tráfico desconocido a Router3   
3. Router3 envía todo el trádico desconocido a Router1   
  
router1:~# route add default gw 200.0.0.254 eth1  
router2:~# route add default gw 200.200.0.254 eth1  
router3:~# route add default gw 200.100.0.254. eth0  
pc1:~# route add default gw 192.168.1.1 eth0  
pc2:~# route add default gw 192.168.2.1 eth0  
pc3:~# route add default gw 192.168.3.1 eth0  
  
**pc1:~# traceroute 192.168.2.2**  
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 40 byte packets  
1 192.168.1.1 (192.168.1.1) 0.160 ms 0.445 ms 0.115 ms  
2 200.200.0.1 (200.200.0.1) 0.263 ms 0.405 ms 0.200 ms  
3 200.100.0.1 (200.100.0.1) 0.205 ms 0.409 ms 0.181 ms  
4 192.168.2.2 (192.168.2.2) 0.342 ms 0.229 ms 0.200 ms

porque pasa por la red 200.100.0.1 ? y la red 200.0.0.0 ?  
  
**pc1:~# ping -nR 192.168.2.2**  
PING 192.168.2.2 (192.168.2.2) 56(124) bytes of data.  
64 bytes from 192.168.2.2: icmp\_seq=1 ttl=62 time=0.425 ms  
RR: 192.168.1.2  
 200.0.0.1  
 200.200.0.1  
 192.168.2.1  
 192.168.2.2  
 192.168.2.2  
 200.100.0.1  
 192.168.1.1  
 192.168.2.2  
64 bytes from 192.168.2.2: icmp\_seq=2 ttl=62 time=0.291 ms (same route)  
…  
Ahi si hace bien la ruta, de ida y vuelta  
6. 4. Mientras realiza ping desde una PC, capture paquetes en un router intermedio y verifique qué paquetes pasan por la interfaz. Por ejemplo, mientras PC1 corre el comando ping a la IP de PC2, analice los paquetes que se visualizan en eth0 y en eth1 de Router3. La captura de paquetes puede hacerse con el comando tcpdump -i interfaz. Por ejemplo: tcpdump -i eth0   
  
 5. En base al punto anterior, ¿qué puede deducir?   
 Como esta configurada la red los paquetes de vuelta (echo reply) vuelven por la interfaz eth0 del router 3 (es decir no vuelven por el mismo camino por el que vienen). La red esta configurada como “circular”.  
  
27)